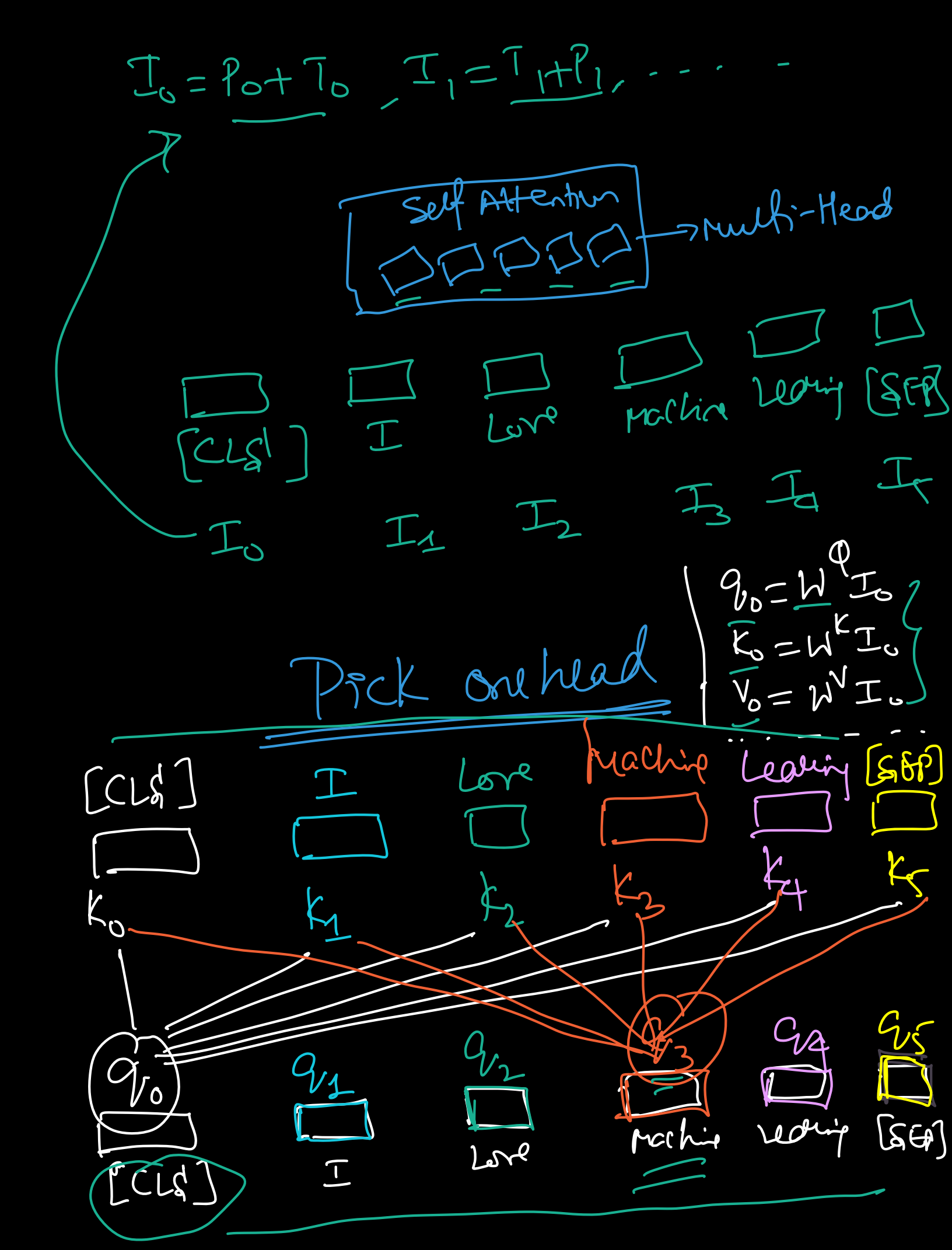
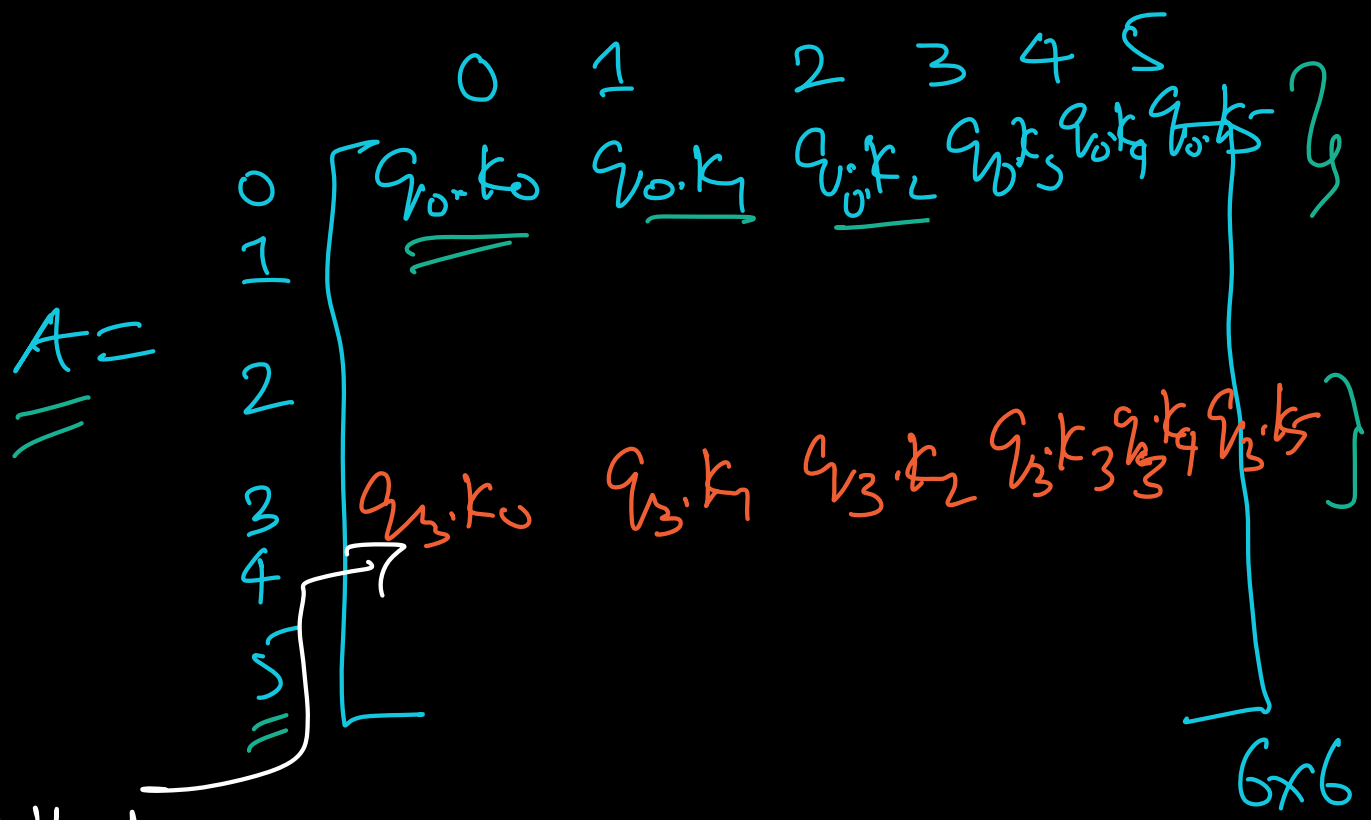

I Love Machine Learning
Separator Token
Start Token
[CLS] I Love Machine Learning [SEP]
Learn
#ing
Further split
Token Embedding
768x1 - BertBase
1536x1 - BertLarge
Embed. vector
Position Embedding
Input Embedding
Capture sequence info
Self-Attention
Q - query
K - key
V - value
Weight Matrices for query, key, value
Multi-Head Self-Attention
Assume 12 heads
Head Embed dim = 768/12 = 64
64x768
64x1
Love


$I_0 = P_0 + T_0$, $I_1 = T_1 + P_1$, ...
Self Attention
multi-Head
[CLS]
I
Love
machine
learning
[SEP]
I_0
I_1
I_2
I_3
I_4
I_5
$q_0 = W^Q I_0$
$k_0 = W^K I_0$
$v_0 = W^V I_0$
Pick one head
[CLS]
I
Love
machine
learning
[SEP]
k_0
k_1
k_2
k_3
k_4
k_5
q_0
q_1
q_2
q_3
q_4
q_5
[CLS]
I
Love
machine
learning
[SEP]



Attention is captured by "dot product"

In general $A_{n \times n}$ matrix

what is n?

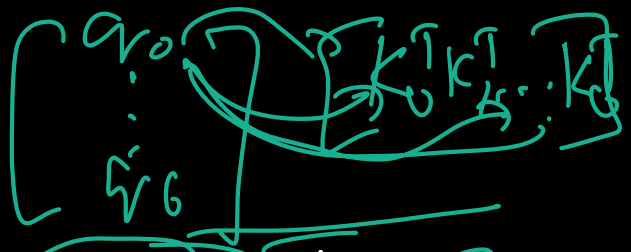
n = #tokens

Let

$$Q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ \vdots \\ q_6 \end{bmatrix}_{6 \times 64} \qquad K = \begin{bmatrix} k_0 \\ k_1 \\ k_2 \\ \vdots \\ k_6 \end{bmatrix}$$

Is q_0 a row vector or column vector?

Notice that:



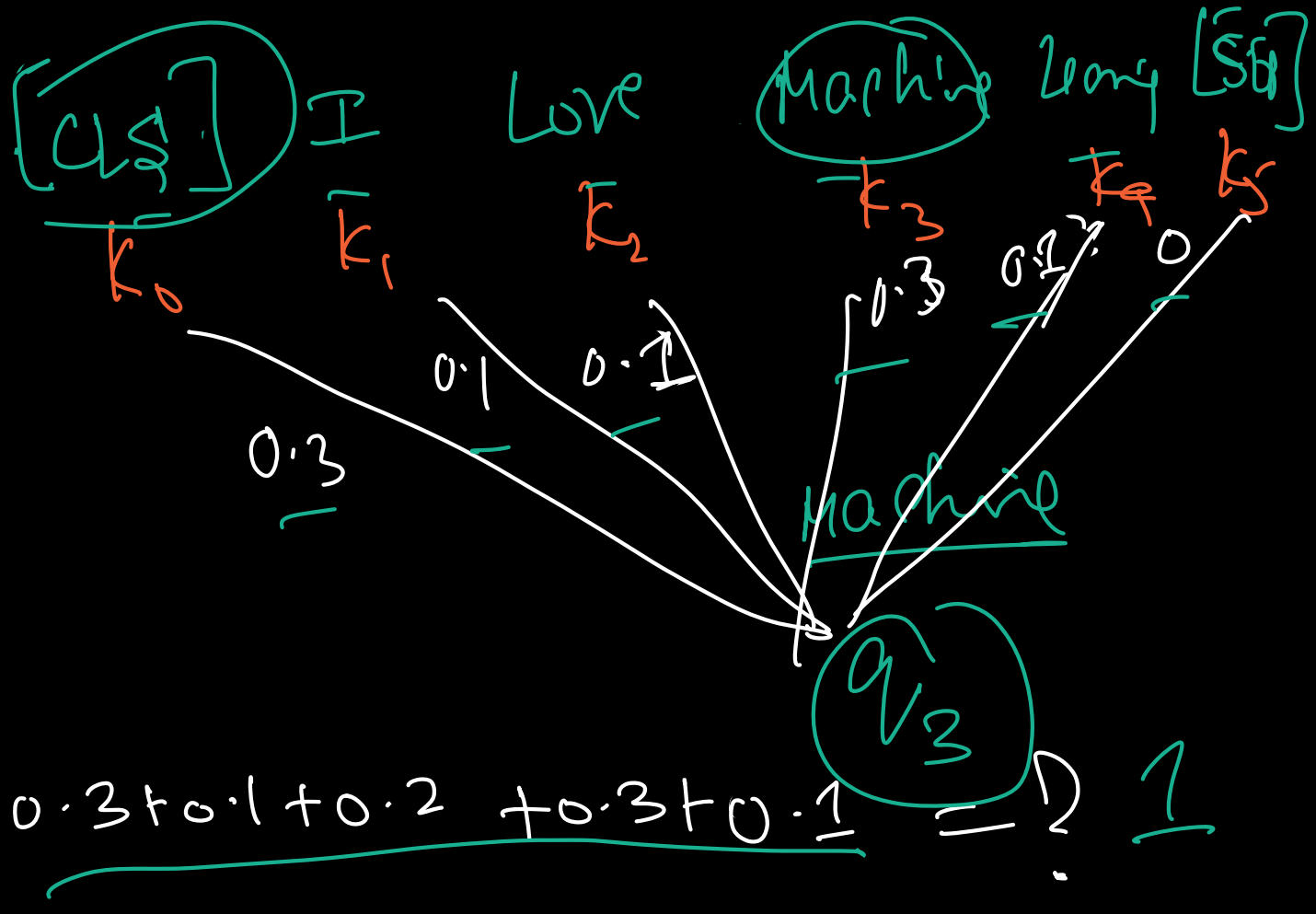
Attention (un-Scaled) $= QK^T$!

Dot product $q_0 \cdot k_0 =$ [row vector] [column vector]

Scaled Dot-Product self-Attention

$$A = \frac{QK^T}{\sqrt{d}} = QK^T/\sqrt{64} = QK^T/8$$

Convert A → Probabilistic Attention

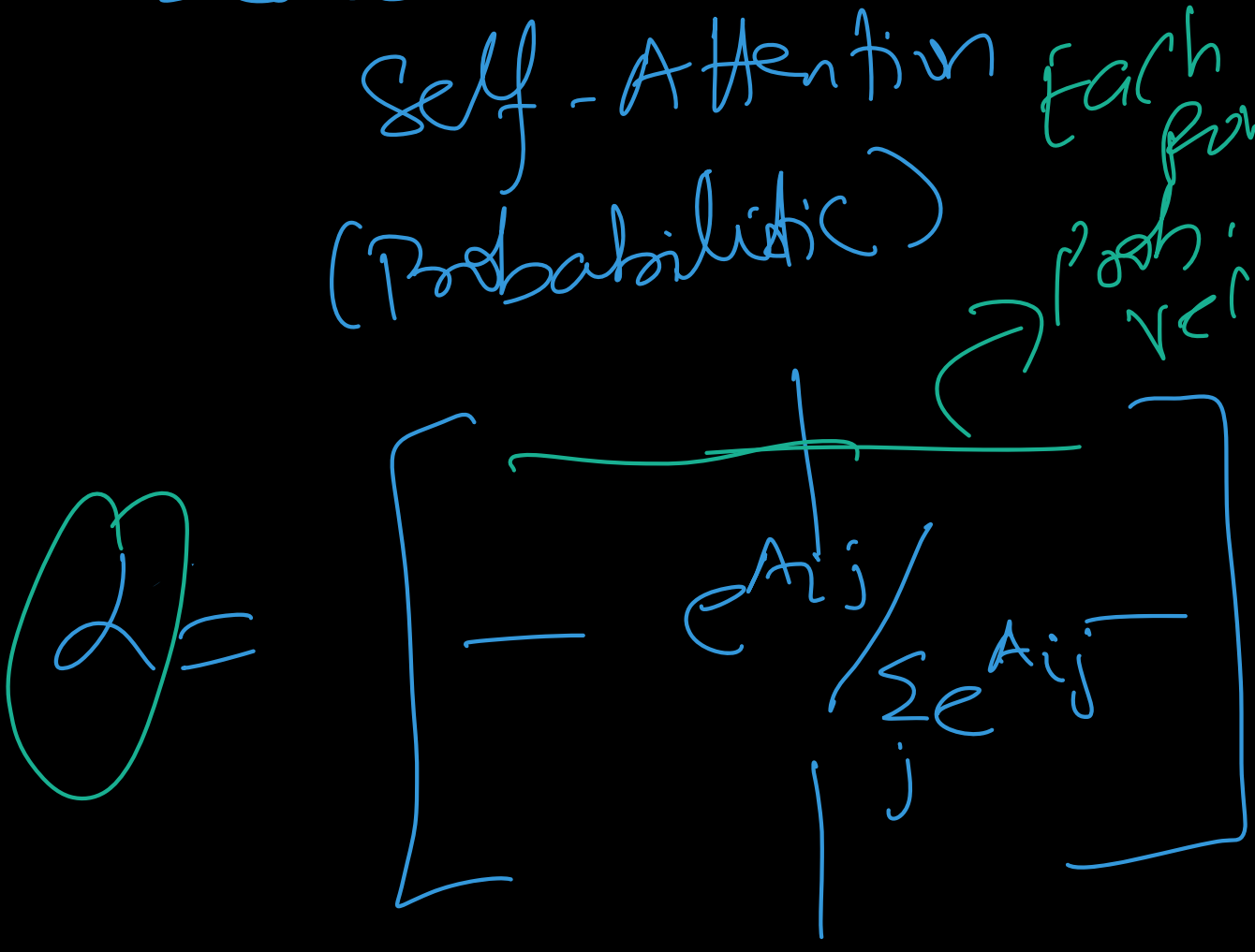


$0.3+0.1+0.2+0.3+0.1 = ? 1$

Softmax

$$\alpha_{ij} = \frac{e^{A_{ij}}}{\sum_j e^{A_{ij}}}$$

Scaled Dot-Product Self-Attention (Probabilistic)



Interpretation:-

How much should machine place emphasis on the 6 tokens [SEP] I love machine learning [CLS]

It is decided by the vector in d.

Which vector in d for machine?

d_3 c-index

d_4 i-index

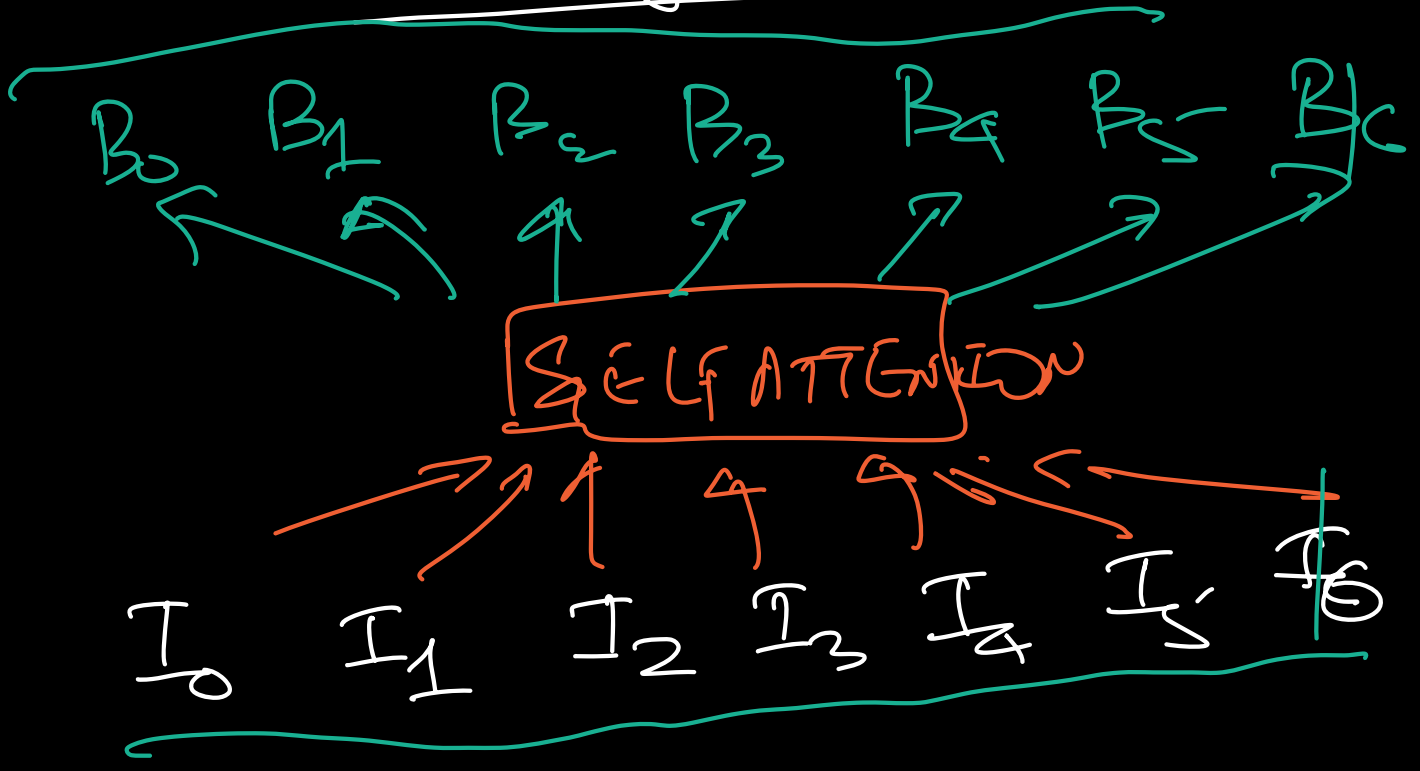
Query, key, values

Query - what you query about

Key - what you match with

Value - New understanding of each token.

output of self-Attention



$B_i \rightarrow$ Blended understanding of all tokens for token i

$\rightarrow V_{[CLS]} = W_V I_{[CLS]}$

So $B_{[Machine]} = \alpha_{30} V_{[CLS]} + \alpha_{31} V_I + \alpha_{32} V_{Love} + \alpha_{33} V_{Machine} + \alpha_{34} V_{learning} + \alpha_{35} V_{[SEP]}$

Concatenate

B_0 B_1 B_2 B_3 B_4 B_5 B_6

64x1 64x1

For one head

Concatenate across heads

B_0^{Concat} = $\begin{bmatrix} B_0^{h_1} \\ B_0^{h_2} \\ \vdots \\ B_0^{h_8} \end{bmatrix}$ 768x1

column vectors

B_0^{Concat} B_1^{Concat} ... B_5^{Concat}

768x1

output projection

$Z_{attention} = B_{Concat}^{T} W_0$

output projection

B_{Concat}: 6x768

W_0: 768x768

6x768

$\begin{bmatrix} I_0 \\ I_1 \\ \vdots \\ I_5 \end{bmatrix}$ 6x768

Input embedding

Remember

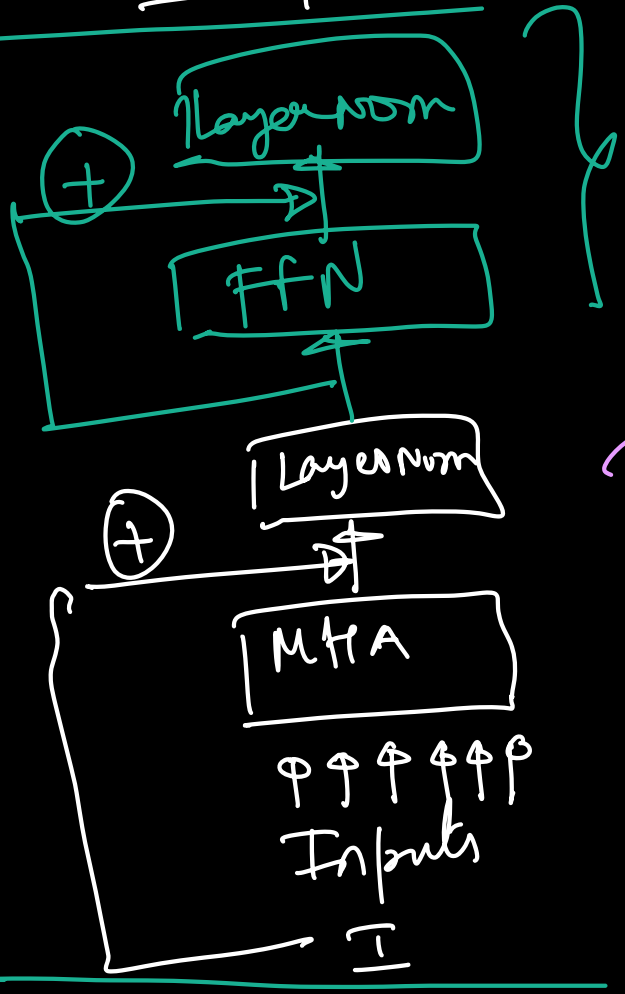
→ Input & output dim of embeddings out of the self-attention is the same!

We are not done yet!

$$\tilde{X} = \text{LayerNorm}\left(I^{(0)} + Z_{\text{attn}}\right)$$

- LayerNorm → For Training Stability
- $I^{(0)}$ → Adding residual connection ensures prev. embedding is still relevant!

Big picture



Layer 0

We just went through this component of one layer of BERT Transformer!

FFN

→ Each token undergoes a non-linear transformation independent of other tokens!!

1. Linear layer expansion

$$H = \underset{6\times 768}{X} \; \underset{768\times 3072}{W_1} + \underset{\text{MHA}}{b_1} \rightarrow 6\times 3072$$

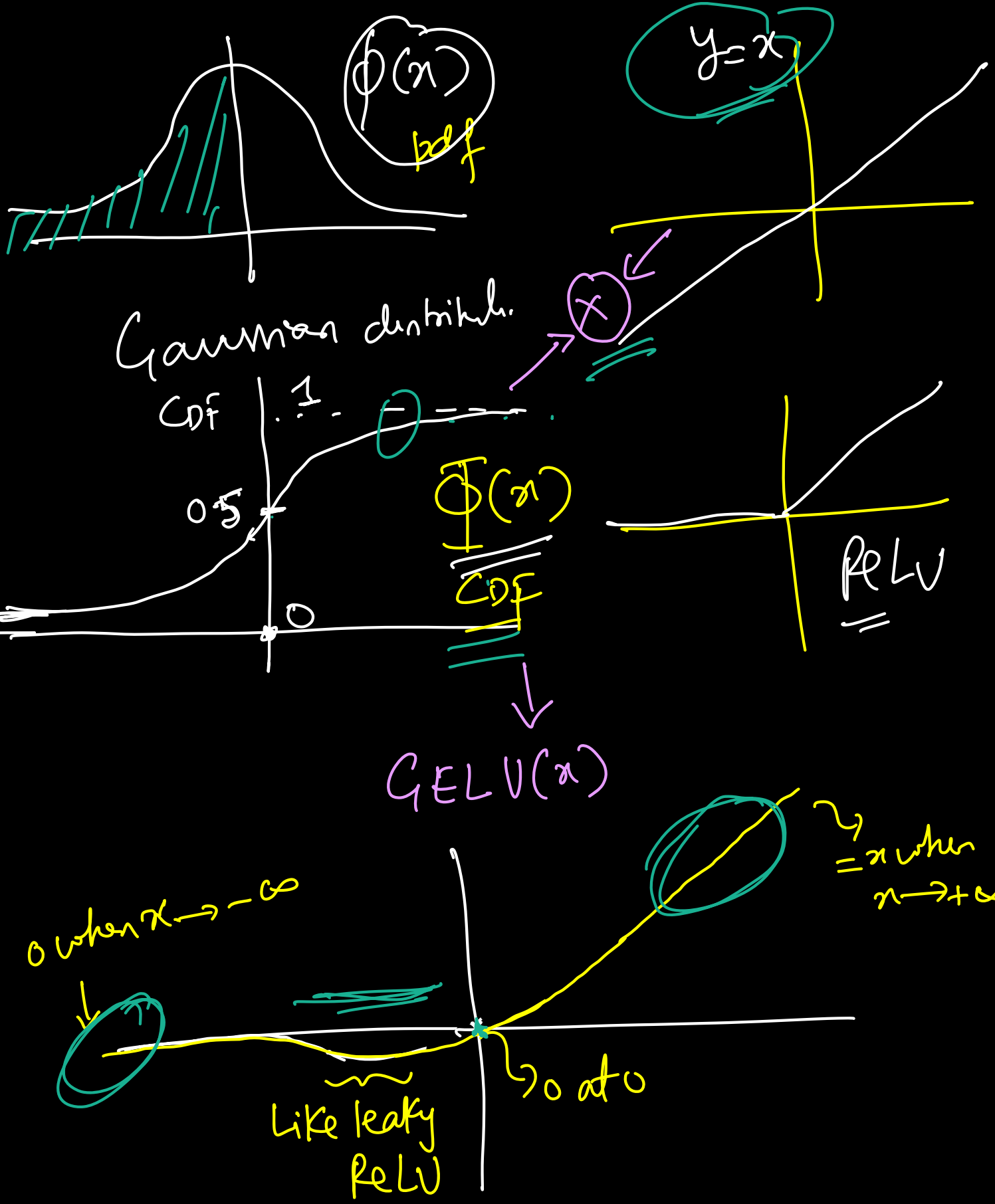
→ Dimension?

↳ expansion

2. GELU activation → Key non-Linearity (Deep understanding)

$$GELU(x) = x \cdot \Phi(x)$$

↳ CDF of Gaussian


Φ(x)
pdf
y=x
Gaussian distribution
CDF
1
0.5
0
Φ(x)
CDF
ReLU
GELU(x)
0 when x → -∞
= x when x → +∞
Like leaky ReLU
0 at 0

b. GELU Activation

$$F = \text{GELU}(H)\, W_2 + b_2$$

- F: 6 × 768
- H: 6 × 3072
- W_2: 3072 × 768
- b_2: 6 × 768
- Compress back

b. Residual + Layer Norm

$$x^{(1)} = \text{LayerNorm}(\tilde{x} + F)$$

- $x^{(1)}$: 6 × 768
- $\tilde{x}$: Prev. Info.
- F: New Info.

→ Same dimension as input to the layer!